PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 9/46 — A3

(11) International Publication Number: **WO 00/38060**

(43) International Publication Date: 29 June 2000 (29.06.00)

(21) International Application Number: PCT/EP99/10170

(22) International Filing Date: 16 December 1999 (16.12.99)

(30) Priority Data:
09/218,551 22 December 1998 (22.12.98) US
09/273,938 22 March 1999 (22.03.99) US

(71) Applicant: KONINKLIJKE PHILIPS ELECTRONICS N.V. [NL/NL]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL).

(72) Inventors: SAVILLE, Winthrop, L.; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL). ROSS, Kevin; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).

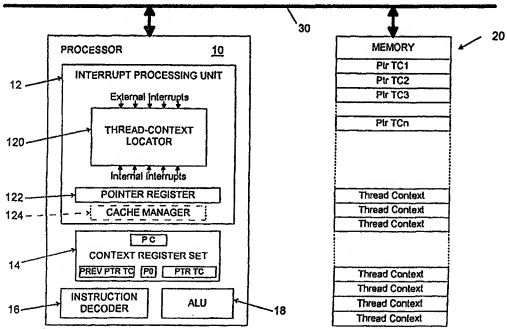
(74) Agent: FAESSEN, Louis, Marie, Hubertus; Internationaal Octrooibureau B.V., Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).

(81) Designated States: CN, JP, KR, SG, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(88) Date of publication of the international search report:
26 October 2000 (26.10.00)

(54) Title: INTERRUPT/SOFTWARE-CONTROLLED THREAD PROCESSING



(57) Abstract

Rapid thread processing is achieved by transferring complete thread contexts between a memory and a context register set. Each thread context is read from a respective memory location in response to either a designated interrupt or an instruction.

(19)日本国特許庁（ＪＰ）　(12) 公表特許公報（Ａ）　(11)特許出願公表番号
特表2002－533807
（P2002－533807A）
(43)公表日　平成14年10月8日(2002.10.8)

(51)Int.Cl.[7]	識別記号	ＦＩ	テーマコード（参考）
Ｇ０６Ｆ　9/46	３１１	Ｇ０６Ｆ　9/46	３１１Ａ　５Ｂ０９８
	３１３		３１３Ｚ
	３４０		３４０Ｂ

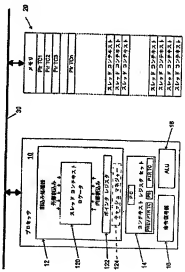
審査請求　未請求　予備審査請求　未請求(全 33 頁)

(21)出願番号　特願2000－590052(P2000－590052)
(86)（22)出願日　平成11年12月16日(1999.12.16)
(85)翻訳文提出日　平成12年８月22日(2000.8.22)
(86)国際出願番号　ＰＣＴ／ＥＰ９９／１０１７０
(87)国際公開番号　ＷＯ００／３８０６０
(87)国際公開日　平成12年６月29日(2000.6.29)
(31)優先権主張番号　０９／２１８，５５１
(32)優先日　平成10年12月22日(1998.12.22)
(33)優先権主張国　米国（ＵＳ）
(31)優先権主張番号　０９／２７３，９３８
(32)優先日　平成11年３月22日(1999.3.22)
(33)優先権主張国　米国（ＵＳ）

(71)出願人　コーニンクレッカ　フィリップス　エレクトロニクス　エヌ　ヴィ
Ｋｏｎｉｎｋｌｉｊｋｅ　Ｐｈｉｌｉｐｓ　Ｅｌｅｃｔｒｏｎｉｃｓ　Ｎ．Ｖ．
オランダ国　5621　ベーアー　アインドーフェン　フルーネヴァウツウェッハ　1
(72)発明者　ウィンスロップ、エル．サビル
オランダ国5656、アーアー、アインドーフェン、プロフ．ホルストラーン、6
(72)発明者　ケビン、ロス
オランダ国5656、アーアー、アインドーフェン、プロフ．ホルストラーン、6
(74)代理人　弁理士　佐藤　一雄　（外３名）
最終頁に続く

(54)【発明の名称】　割込み／ソフトウエア制御スレッド処理

(57)【要約】
【解決手段】　メモリとコンテキスト・レジスタ・セットとの間で完全なスレッド・コンテキストを転送することによって迅速なスレッド処理が行われる。各スレッド・コンテキストは、指定された割込みまたは命令に応答してそれぞれのメモリ位置から読出される。



【特許請求の範囲】

【請求項1】

コンテキスト・レジスタ・セットへのそれぞれのスレッド・コンテキストの読み込みに応答して、複数の異なるスレッドのうちのいずれかを選択的に実行するために単一のプロセッサを動作する方法において、

a．複数の割込みのそれぞに、それぞれのメモリ位置を関連付けることと、

b．複数のスレッド・コンテキスト・ポインタのぞれそれを、前記メモリ位置に保存し、このポインタのそれぞれは、前記スレッドのうちの1つのスレッドの動作状態を十分に記述するスレッド・コンテキストを含むためのメモリ位置を特定することと、

c．いずれかの割込みの発生に応答して、前記割込みに関連付けられているそれぞれのメモリ位置からポインタを読出すことと、

d．スレッド・コンテキスト・ポインタ読出しによって特定されるメモリ位置から、スレッド・コンテキストをコンテキスト・レジスタ・セットに読込むことと、

e．スレッド・コンテキスト読出しによって記述されているスレッドを実行することと、

を備える単一のプロセッサを動作する方法。

【請求項2】

請求項1記載の方法であって、スレッド・コンテキスト・ポインタのそれぞれが、異なるメモリ位置を特定する方法。

【請求項3】

請求項1記載の方法であって、1より多いスレッド・コンテキスト・ポインタが、同じメモリ位置を特定する方法。

【請求項4】

請求項1記載の方法であって、プロセッサにより実行されているプログラム・ストリーム中における割込み命令の1つ又は複数のスレッド変更命令の1つに応答して実行しているスレッドを、プロセッサが変更する方法。

【請求項5】

請求項４記載の方法であって、スレッド変更命令は、現在実行しているスレッドのコンテキストを保存するため、及び、実行されているスレッドを異なるスレッドに変更するための、ＳＷＩＴＣＨ命令を含む方法。

【請求項６】

請求項５記載の方法であって、

プロセッサは、先に実行したスレッドに対するコンテキストのアドレスを保存し、

前記スレッド変更命令は、実行しているスレッドを、前記先に実行したスレッドに変更するための第１のＳＷＩＴＣＨ命令を含む方法。

【請求項７】

請求項５記載の方法であって、第２のＳＷＩＴＣＨ命令を含み、この第２のＳＷＩＴＣＨ命令は、第２のＳＷＩＴＣＨ命令により特定されているメモリ位置に含まれているそのコンテキストを有するスレッドに、実行されているスレッドを変更するためのものである方法。

【請求項８】

請求項７記載の方法であって、第２のＳＷＩＴＣＨ命令は、前記コンテキストのメモリ位置を暗黙的に特定する方法。

【請求項９】

請求項７記載の方法であって、第２のＳＷＩＴＣＨ命令は、前記コンテキストのメモリ位置を明示的に特定する方法。

【請求項１０】

請求項９記載の方法であって、第２のＳＷＩＴＣＨ命令は、前記コンテキストのメモリ位置を特定するポインタを含む方法。

【請求項１１】

請求項９記載の方法であって、第２のＳＷＩＴＣＨ命令は、前記コンテキストのメモリ位置を特定するポインタを含むメモリ位置を特定する方法。

【請求項１２】

請求項４記載の方法であって、スレッド変更命令は、現在実行しているスレッドのコンテキストを保存することなく、異なるスレッドを実行するためのＲＥＳ

ＴＯＲＥ命令を含む方法。

【請求項１３】

請求項１２記載の方法であって、

プロセッサは先に実行したスレッドに対するコンテキストのアドレスを保存し、

前記スレッド変更命令は、実行されているスレッドを前記先に実行したスレッドに変更するための第１のＲＥＳＴＯＲＥ命令を含む方法。

【請求項１４】

請求項１２記載の方法であって、第２のＲＥＳＴＯＲＥ命令を含み、この第２のＲＥＳＴＯＲＥ命令は、第２のＲＥＳＴＯＲＥ命令により特定されているメモリ位置に含まれているそのコンテキストを有するスレッドに、実行されているスレッドを、変更するためのものである方法。

【請求項１５】

請求項１４記載の方法であって、第２のＲＥＳＴＯＲＥ命令は前記コンテキストのメモリ位置を暗黙的に特定する方法。

【請求項１６】

請求項１４記載の方法であって、第２のＲＥＳＴＯＲＥ命令は前記コンテキストのメモリ位置を明示的に特定する方法。

【請求項１７】

請求項１６記載の方法であって、第２のＲＥＳＴＯＲＥ命令は前記コンテキストのメモリ位置を特定するポインタを含む方法。

【請求項１８】

請求項１６記載の方法であって、第２のＲＥＳＴＯＲＥ命令は前記コンテキストのメモリ位置を特定するポインタを含むメモリ位置を特定する方法。

【請求項１９】

請求項４記載の方法であって、スレッド変更命令は、割込みをエミュレートするためのＳ／Ｗ　ＩＮＴＥＲＲＵＰＴ命令を含む方法。

【請求項２０】

請求項４記載の方法であって、複数のコンテキスト・レジスタ・セットが設け

られ、前記単一のプロセッサは前記コンテキスト・レジスタ・セットの所定の１つに含まれている方法。

【請求項２１】

請求項１記載の方法であって、コンテキスト・レジスタ・セットは、

ａ．プロセッサにより現在実行されているスレッドのコンテキストのメモリ・アドレスを保存するためのレジスタと、

ｂ．スレッド・コンテキスト・ポインタによって特定されていないスレッド・コンテキストの位置を特定するために用いられるレジスタと、

を含む方法。

【請求項２２】

請求項１記載の方法であって、コンテキスト・レジスタ・セットはプロセッサで先に実行されたスレッドのコンテキストのメモリ・アドレスを保持するためのレジスタを含む方法。

【請求項２３】

請求項１記載の方法であって、プロセッサは、複数のコンテキスト・レジスタ・セットを含む方法。

【請求項２４】

請求項２３記載の方法であって、コンテキスト・レジスタ・セットの数は、プロセッサで実行すべきスレッドの数より、少ない方法。

【請求項２５】

請求項２４記載の方法であって、スレッド・コンテキストが読み込まれるコンテキスト・レジスタ・セットを選択するために、優先順位を基にしたアルゴリズムが採用される方法。

【請求項２６】

請求項２４記載の方法であって、

スレッドはより高い優先順位のスレッド及びより低い優先順位のスレッドを含み、

より低い優先順位のスレッドよりも、より高い優先順位のスレッドが利用可能なように、比例して大きい数のコンテキスト・レジスタ・セットが形成される、

される方法。

【請求項２７】

請求項２６記載の方法であって、少なくとも１つのスレッドの優先順位を、プロセッサの動作中に変更できる方法。

【請求項２８】

請求項２６記載の方法であって、

より高い優先順位のスレッド又はより低い優先順位のスレッドの１つのためのスレッド・コンテキストの読出し時に、セット選択アルゴリズムが用いられ、

それぞれの優先順位で使用可能に形成されたコンテキスト・レジスタ・セットのすべてが、読出されているもの以外のスレッドのコンテキストを既に含んでいる方法。

【請求項２９】

請求項２８記載の方法であって、前記アルゴリズムは、前記スレッド・コンテキストを、最後に保存されたスレッド・コンテキストを含んでいる使用可能なコンテキスト・レジスタに、読み込む方法。

【請求項３０】

コンテキスト・レジスタ・セットへのそれぞれのスレッド・コンテキストの読み込みに応答して、複数の異なるスレッドのうちのいずれかを、複数のプロセッサのぞれぞれが選択的に実行するようにプロセッサのネットワークを動作する方法において、

ａ．複数の割込みのそれぞれに、それぞれのメモリ位置を関連付けることと、

ｂ．複数のスレッド・コンテキスト・ポインタのそれぞれを、前記メモリ位置に保存し、このポインタのそれぞれは、前記スレッドのうちの１つのスレッドの動作状態を十分に記述するスレッド・コンテキストを含むためのメモリ位置を特定するすることと、

ｃ．いずれかの割込みの発生に応答して、前記割込みに関連させられているそれぞれのメモリ位置からポインタを読出すことと、

ｄ．スレッド・コンテキスト・ポインタ読出しによって特定されるメモリ位置から、スレッド・コンテキストをコンテキスト・レジスタ・セットに読込むこと

と、

ｅ．スレッド・コンテキスト読出しによって記述されているスレッドを実行することと、

を備えるプロセッサのネットワークを動作する方法。

【請求項３１】

請求項３０記載の方法であって、スレッドの少なくとも１つが、１より多いプロセッサにより実行できる共通スレッドである方法。

【発明の詳細な説明】

【０００１】

発明の背景

１．発明の分野

本発明はスレッド指向処理（thread-oriented processing）に関するものであり、更に詳しくいえば、コンテキストの使用による多重スレッド処理（multilpe-thread）に関するものである。

【０００２】

２．関連技術についての説明

多重スレッド処理においては、プロセッサは複数の異なる処理（一般に「スレッド」と呼ばれている）を逐次実行できる。所定の時間期間の経過後または特定の命令の受け取り時などの、特定の事象の発生時に、プロセッサは１つのスレッドの実行を保留し、保留するスレッドの状態を記述しているコンテキストを保存し、別のスレッドの実行を開始する。そのスレッドもそれぞれのコンテキストにより記述されている。各「コンテキスト」は、プロセッサが新しいスレッドの実行を開始するため、または保留されているスレッドの動作を継続するために必要な情報を含んでいる。通常はこの情報はメモリ・アドレスと、ステータス情報と、データとを含んでいる。

【０００３】

保存されているスレッド・コンテキストの迅速なアクセスを直接行うために、いくつかのプロセッサはローカル・コンテキスト・レジスタの多数のバンクを含んでいる。しかし、これは硬直した構成であって、実行できるスレッドの数を一定数に制限する。また、設けられているレジスタ・バンクの数よりスレッドの数が少ない時は常に効率が悪い。

【０００４】

あるいは、多数のコンテキストを別々のメモリに保存できる。そのようなやり方の種々の例が米国特許第５３４９６８０号に記述されている。その特許に従来の情報処理装置として記述されている、それらの例の１つにおいては、主プロセッサは、種々のアプリケーション・プロセスから動作を逐次実行するためのアプ

リケーション・サポート装置と、情報処理装置における動作を制御するためのシステム・サポート装置とを含んでいる。この装置の効率は低いと記述されている。その理由は、アプリケーション・サポート装置とシステム・サポート装置は同時には決して動作させられないからである。米国特許第５３４９６８０号は保存されている多数のコンテキストを利用する別の装置を提案しているが、それらの装置のおのおのは２つの別々のプロセッサを使用することを要する。これは効率を向上するための費用のかかるやり方である。

【０００５】

発明の概要

本発明の目的は、一定数のローカル・コンテキスト・レジスタを使用するという硬直性の不利益をこうむらない、単一プロセッサによる迅速かつ効率的な多重スレッド処理を可能にすることである。

【０００６】

この目的およびその他の目的は、コンテキストをコンテキスト・レジスタ・セットとメモリとの間で交換（swapping）することによって達成される。これによって、最少のハードウェアにより多重スレッド処理が可能にされる。本発明によれば、

・複数の割込みのそれぞれに、それぞれのメモリ位置を関連付けることと、

・複数のスレッド・コンテキスト・ポインタのそれぞれを、前記メモリ位置に保存し、このポインタのそれぞれは、前記スレッドのうちの１つのスレッドの動作状態を十分に記述するスレッド・コンテキストを含むためのメモリ位置を特定することと、

・いずれかの割込みの発生に応答して、前記割込みに関連させられているそれぞれのメモリ位置からポインタを読出すことと、

・スレッド・コンテキスト・ポインタ読出しによって特定されているメモリ位置から、スレッド・コンテキストをコンテキスト・レジスタ・セットに読込むことと、

・スレッド・コンテキスト読出しによって記述されているスレッドを実行することと、

を含む方法が用いられる。

【０００７】

それぞれの割込みに関連させられているメモリ位置に、コンテキスト自体ではなくて、コンテキストのポインタを保存することによって、割込みは特定のコンテキストから関係を断たれる。これによって割込みに対する応答の決定における融通性が高くなる。また、ただ１つのメモリ位置に１より多い割込みに共通のコンテキストを保存し、且つ、この共通コンテキストのアドレスをそれらの割込みのための各ポインタに含めることによって、メモリを節約できる。更に、コンテキスト自体をポインタにより特定されているメモリ位置に直接保存することによって、１つのスレッドから別のスレッドへの迅速なプロセッサ変更が可能にされる。

【０００８】

ここで使用している「メモリ」という語は、それが用いられる態様に一致するものと一般に解釈されるべきであることを意図しており、かつ、制約なしに、レジスタ、ＲＡＭ、ＤＲＡＭ、ＲＯＭ、及び、それらの装置の組合わせなどの、各種の揮発性装置および不揮発性装置を含むことに注目されたい。また、「読出し」は、１つのメモリからの情報の検索、及び、他のメモリへのそれの書込みを意味する。

【０００９】

本発明の特に有利な実施例では、プロセッサによって実行されているプログラム・ストリーム（すなわち、そのプログラムを構成している命令列）中のある命令が、プロセッサにおいてコンテキストの変更を直接行う。これによってスレッド自体による迅速なコンテキスト変更の開始が可能になる。

【００１０】

【発明の実施の形態】

好適な実施例についての説明

図１の多重スレッド処理装置１は、単一のプロセッサ１０とメモリ２０を含んでいる。例示したプロセッサ１０は、プログラム・カウンタ・レジスタにより特定された命令を順次実行するためにクロックパルスを利用する、ハードウェアで

加速された装置（hardware-accelerated unit）である。通常は、プログラム・カウンタ・レジスタは、プロセッサが読出して、実行する次の命令のメモリ位置を含んでいる。

【００１１】

プロセッサは、割込み処理器１２と、コンテキスト・レジスタ・セット１４と、命令復号器１６と、算術論理装置１８とを含む。この実施例では、メモリ２０は、とくに、複数のスレッド・コンテキスト・ポインタＰｔｒ　ＴＣ１、Ｐｔｒ　ＴＣ２、Ｐｔｒ　ＴＣ３、．．．Ｐｔｒ　ＴＣｎと、複数のスレッド・コンテキストを保存するための多数のメモリ位置を有するＲＡＭを備えている。

【００１２】

プロセッサ１０とメモリ２０は、共通バス３０に接続されて相互に通信するとともに、このバスに接続されている他のハードウェアとも通信する。このバスは、アドレス、割込み、データ、読出しストローブ、書込みストローブ、装置選択ストローブなどの情報を伝えるためのそれぞれのラインを含んでいる。好ましくは、このバスは、プロセッサおよびメモリと共通のシリコン基板上に少なくとも部分的に形成された高速バスである。

【００１３】

プロセッサ１０の動作は割込みおよびプログラム・ストリーム中の命令によって全体的（entirely）に制御される。割込みは、バス３０から受けられた外部割込み、またはプロセッサ自体で、たとえば、プロセッサ内のタイマ（図示せず）から、発生された内部割込みとすることができる。各外部割込みには、メモリ２０内の、スレッド・コンテキスト・ポインタＰｔｒ　ＴＣ１、Ｐｔｒ　ＴＣ２、Ｐｔｒ　ＴＣ３、．．．Ｐｔｒ　ＴＣｎの１つが保存されている所定の位置が関連付けられている。それらのポインタのおのおのには、バス３０に接続されているハードウェアにより利用されるスレッド・コンテキストの１つが関連付けられ、特定のコンテキストが保存されているメモリ位置を特定する。１より多いポインタ（たとえば、Ｐｔｒ　ＴＣ１およびＰｔｒ　ＴＣ３）に同じスレッド・コンテキストを関連付けることができ（すなわち、「共用する」）、したがって、同じメモリ位置を特定する。これは、たとえば、バス３０に接続されている１より

多い装置、たとえば、受信ＦＩＦＯおよび送信ＦＩＦＯ、が同じコンテキストを利用するならば、有用である。内部割込みのそれぞれが、メモリ２０内の、プロセッサにより利用されているスレッド・コンテキストの１つが保存されている所定の位置に関連付けられる。

【００１４】

あるコンテキストが共用されているならば、処理時間が長くなる。その理由は、割込みサービス・スレッドが割込みの元を決定しなければならないからである。しかし、より少ないメモリが必要とされる。逆に、ある割込みがあるコンテキストを独占的に使用するものとすると、割込みサービス・スレッドは割込みの元を固有的に特定でき、処理応答時間はより短い。これによって装置のアーキテクチャに融通性が与えられる。

【００１５】

全ての割込みは、割込み処理器１２によって処理される。この割込み処理器はスレッド・コンテキスト・ロケータ１２０とポインタ・レジスタ１２２を含み、かつキャッシュ・マネジャー１２４を含むことを選択できる。スレッド・コンテキスト・ロケータ１２０は、外部割込みおよび内部割込みに関連付けられているメモリ位置を識別するアドレスを生成する。任意の割込みに応答して、スレッド・コンテキスト・ロケータは、スレッド・コンテキスト・ポインタＰｔｒ　ＴＣ１、Ｐｔｒ　ＴＣ２、Ｐｔｒ　ＴＣ３、．．．Ｐｔｒ　ＴＣｎの関連付けられている１つが保存されているメモリ２０の所定の位置を特定するアドレスを生成する。その後で、プロセッサは、スレッド・コンテキスト位置をこのメモリ位置からポインタ・レジスタ１２２に読み込む。外部割込みの例には、ＦＩＦＯ　ＴＲＡＮＳＭＩＴ　ＣＯＭＰＬＥＴＥ、ＦＩＦＯ　ＤＡＴＡ　ＲＥＣＥＩＶＥＤ、ＤＭＡ　ＴＲＡＮＳＦＥＲ　ＣＯＭＰＬＥＴＥ、ＦＩＦＯ　ＴＲＡＮＳＭＩＴ　ＦＡＩＬＵＲＥ等が含まれる。内部割込みの例には、内部タイマ等が含まれる。

【００１６】

好ましくは、スレッド・コンテキスト・ロケータは、プログラム可能なルックアップ・テーブルまたは符号器などのアドレスを発生するための専用の優先順位感知ハードウェアを備えており、両者ともこの技術で周知である。両者ともアド

レッシングの速さを最速化し、かつ、より低い優先順位の割込みより先により高い優先順位の割込みの処理を可能にする。

【００１７】

コンテキスト・レジスタ・セット１４は、現在プロセッサにより実行されているスレッドのコンテキストを含む複数のレジスタを備えている。好適な実施例では、レジスタ・セット１４は、以下を含んでいる。

【００１８】

・　現在プロセッサにより実行されているスレッドのためのコンテキストのメモリ・アドレスを保存するためのレジスタＰＴＲ　ＴＣ。

【００１９】

・　先に（previously）プロセッサにより実行されたスレッドのためのコンテキストのメモリ・アドレスを保存するためのレジスタＰＲＥＶ　ＰＴＲ　ＴＣ。

【００２０】

・　メモリ２０に保存されているスレッド・コンテキスト・ポインタにより特定されないことがあるスレッド・コンテキストの位置を特定するために用いられるレジスタＰ０。

【００２１】

・　メモリ２０内で次にアクセスすべき命令のアドレスを特定するために、継続的に更新するプログラム・カウンタ・レジスタＰＣ。

【００２２】

・　１つまたは複数の汎用レジスタ（図示せず）。

【００２３】

・　メモリ２０から読み出され、又は、算術論理装置１８で発生するデータを含むための１つまたは複数のデータ・レジスタ（図示せず）。

【００２４】

命令復号器１６は、メモリ２０から読出された命令を、算術論理装置１８によって処理すべきより低いレベルのオペレーション・コードに変換するための、シーケンサまたはマイクロシーケンサなどの、従来のハードウェア部品である。算術論理装置も従来のハードウェア部品である。

【００２５】

図２は、順次発生する、複数の割込み例とコンテキスト変更のための複数の命令の例との制御の下にあるプロセッサ１０の動作を示す。ボックス４０はそれらの割込みおよびコンテキスト変更命令が起きるにつれてのプロセッサ内の事象列を表す。表されているコンテキスト変更命令の例（ＲＥＳＴＯＲＥ　ＰＲＥＶ　ＴＣ、ＳＷＩＴＣＨ　Ｐ０　ＴＯ　ＴＣ、ＲＥＳＴＯＲＥ　ＦＭ　Ｐ０　ＴＣ、ＳＷＩＴＣＨ　ＴＯ　ＰＲＥＶ　ＴＣ、及び、Ｓ／Ｗ　ＩＮＴＥＲＲＵＰＴ）がコンテキスト・レジスタ・セット１４における変更を行う。これについては後で詳しく説明する。

【００２６】

ボックス２０′は、特定のポインタと、それらのポインタによって位置付けされたスレッド・コンテキストと、シーケンス４０において発生する命令ＳＷＩＴＣＨ　ＴＯ　Ｐ０　ＴＣ及びＲＥＳＴＯＲＥ　ＦＭ　Ｐ０　ＴＣにより位置決めされたスレッド・コンテキストとを、含んでいる、メモリ２０内の位置を表している。この特定の例では、メモリ位置２０′は、以下を含んでいる。

【００２７】

・　割込みＩＮＴ　１に応答してプロセッサにより読出され、かつ関連させられているスレッド・コンテキストＴＨＲＥＡＤ　ＣＯＮＴＥＸＴ　Ｆのメモリ位置を含むポインタＰｔｒ　ＴＣ　ＩＮＴ　１。

【００２８】

・　割込みＥＸＴ　１に応答してプロセッサにより読出され、かつ関連させられているスレッド・コンテキストＴＨＲＥＡＤ　ＣＯＮＴＥＸＴ　Ａのメモリ位置を含むポインタＰｔｒ　ＴＣ　ＥＸＴ　１。

【００２９】

・　割込みＥＸＴ　２に応答してプロセッサにより読出され、かつ関連させられているスレッド・コンテキストＴＨＲＥＡＤ　ＣＯＮＴＥＸＴ　Ｂのメモリ位置を含むポインタＰｔｒ　ＴＣ　ＥＸＴ　２。

【００３０】

・　割込みＥＸＴ　３に応答してプロセッサにより読出され、かつ関連させられ

ているスレッド・コンテキストTHREAD CONTEXT Cのメモリ位置を含むポインタPtr TC EXT 3。

【0031】

・ スレッド・コンテキストTHREAD CONTEXT A。

【0032】

・ スレッド・コンテキストTHREAD CONTEXT B。

【0033】

・ スレッド・コンテキストTHREAD CONTEXT C。

【0034】

・ スレッド・コンテキストTHREAD CONTEXT D。

【0035】

・ スレッド・コンテキストTHREAD CONTEXT E。

【0036】

・ スレッド・コンテキストTHREAD CONTEXT F。

【0037】

この実施例では、命令RESTORE PREV TC、SWITCH TO P0 TC、RESTORE FM P0 TC、SWITCH TO PREV TCおよびS/W INTERRUPTは、メモリ20に含まれており、それらのメモリ位置がプログラム・カウンタ・レジスタPCにより特定されるたびに、これらの命令が命令復号器16に読み込まれる。

【0038】

図2に示されているシーケンスは時刻t0で始まる。その時にはプロセッサ10は、コンテキスト・レジスタ・セットに含まれているTHREAD CONTEXT Aにより表されているスレッドAを実行している。簡単にするために、各割込みは可能にされていて、それが発生されると直ちに実行されると仮定することにする。

【0039】

時刻t1では、プロセッサ10は、バス30内の割込みラインの1つを介して外部割込みEXT 2を受ける。この割込みに応答して、プロセッサは、以下の

処理を行う。

【００４０】

・ コンテキスト・レジスタ・セット１４に現在含まれているスレッド・コンテキスト、すなわち、スレッドAに対するコンテキスト（ＴＨＲＥＡＤ ＣＯＮＴＥＸＴ Ａ）を、コンテキスト・レジスタ・セット１４内のＰＴＲ ＴＣレジスタに含まれている現在のスレッド・コンテキスト・アドレスにより特定されているメモリ位置に読み込む。

【００４１】

・ レジスタＰＴＲ ＴＣに現在含まれているアドレスをレジスタＰＲＥＶ ＰＴＲ ＴＣに読み込む。

【００４２】

・ 外部割込みＥＸＴ ２に関連付けられているメモリ位置、すなわち、ポインタＰｔｒ ＴＣ ＥＸＴ ２が保存されているメモリ位置、を特定するアドレスを、スレッド・コンテキスト・ロケータ１２０に発生させる。

【００４３】

・ ＴＨＲＥＡＤ ＣＯＮＴＥＸＴ Ｂのメモリ位置を特定しているポインタＰｔｒ ＴＣ ＥＸＴ ２を、ポインタ・レジスタ１２２に読み込む。

【００４４】

・ このスレッド・コンテキストをコンテキスト・レジスタ・セット１４に読み込み、スレッドＢの実行を開始する。

【００４５】

時刻ｔ２では、プロセッサ１０は、スレッドＢ内の最後の命令として、命令ＲＥＳＴＯＲＥ ＰＲＥＶ ＴＣに出くわす。この命令の実行において、プロセッサは次の処理を実行する。

【００４６】

・ ＰＲＥＶ ＰＴＲ ＴＣレジスタからのアドレスをポインタ・レジスタ１２２に読み込む。

【００４７】

・ 所望により、ＰＴＲ ＴＣレジスタからのアドレスをＰＲＥＶ ＰＴＲ Ｔ

Cレジスタに読み込む。

【0048】

・ ポインタ・レジスタ122によって特定されたメモリ20′位置からのスレッド・コンテキストAを、コンテキスト・レジスタ・セット14に読み込み、スレッドAの実行を再び開始する。

【0049】

時刻t3では、スレッドAを実行している間に、プロセッサは、プログラム・カウンタ・レジスタPCにより特定されたメモリ20′内のそれぞれの位置から命令SWITCH TO P0 TCを読出す。この命令は1つのスレッドから、メモリ20′に予め保存されているスレッド・コンテキスト・ポインタPtr TC EXT1、Ptr TC EXT2または、Ptr TC EXT3のいずれによっても特定されないことがある別のスレッドへの切換えを可能にするために与えられる。この切換えを容易にするために、SWITCH TO P0 TC命令に先行する命令の1つが、切換えるべきスレッドのためのコンテキストの位置を特定するアドレスを、コンテキスト・レジスタ・セット内のP0レジスタに予め保持する。この例では、切換えは、現在のスレッドAからメモリ20′内のどこかに配置されているスレッドDへである。この命令の実行においては、プロセッサは、以下の処理を実行する。

【0050】

・ THREAD CONTEXT Aを、コンテキスト・レジスタ・セット14内のレジスタPTR TCに含まれている現在のスレッド・コンテキスト・アドレスによって特定されているメモリ位置に読み込む。

【0051】

・ レジスタPTR TCに現在含まれているアドレスをレジスタPREV PTR TCに読み込む。

【0052】

・ P0レジスタからのアドレスをポインタ・レジスタ122に読み込む、

・ ポインタ・レジスタ122により特定されているメモリ20′内の位置からのスレッド・コンテキスト（THREAD CONTEXT D）を、コンテキ

スト・レジスタ・セット１４に読み込む。

【００５３】

・ スレッドDの実行を開始する。

【００５４】

時刻ｔ４では、スレッドDの実行中にメモリ２０′から読出された命令に応答して、プロセッサ１０は命令ＲＥＳＴＯＲＥ ＦＭ Ｐ０ ＴＣをメモリ２０′内のそれぞれの位置から読出すことによってその命令を実行する。命令ＲＥＳＴＯＲＥ ＦＭ Ｐ０ ＴＣは、命令ＳＷＩＴＣＨ ＴＯ Ｐ０ ＴＣに類似するが、コンテキスト・レジスタ・セット１４に現在のコンテキストを保存することなくスレッドのコンテキストを読出す。この命令の実行を容易にするために、ＲＥＳＴＯＲＥ ＦＭ Ｐ０ ＴＣ命令に先行する命令の１つが、復帰すべきスレッドに対するコンテキストの位置を特定するアドレスをＰ０レジスタに予め保存する。この例では、プロセッサは、以下の処理によりスレッドＥを復帰する。

【００５５】

・ Ｐ０レジスタからのＴＨＲＥＡＤ ＣＯＮＴＥＸＴ Ｅに対するアドレスをポインタ・レジスタに読み込む。

【００５６】

・ 所望により、ＰＴＲ ＴＣレジスタからのアドレスを、ＰＲＥＶ ＰＴＲ ＴＣレジスタに読み込む。

【００５７】

・ メモリ２０′からのＴＨＲＥＡＤ ＣＯＮＴＥＸＴ Ｅを、コンテキスト・レジスタ・セットに読み込む。

【００５８】

時刻ｔ５では、スレッドＥの実行中に、プロセッサは、プログラム・カウンタ・レジスタＰＣにより特定されているメモリ２０′内のそれぞれの位置から命令ＳＷＩＴＣＨ ＴＯ Ｐ０ ＴＣを読み出す。この命令は、現在実行中のスレッドから、先に実行されていたスレッドへの切換えを可能にするために与えられる。この例では、切換えは現在のスレッドＥからスレッドDへである。この命令の実行に際しては、プロセッサは、以下の処理を行う。

【００５９】

・ ＴＨＲＥＡＤ ＣＯＮＴＥＸＴ Ｅを、コンテキスト・レジスタ・セット１４内のレジスタＰＴＲ ＴＣに含まれている現在のスレッド・コンテキスト・アドレスによって特定されるメモリ２０′ 内のメモリ位置に読み込む。

【００６０】

・ ＰＲＥＶ ＰＴＲ ＴＣレジスタからのアドレスを、ポインタレジスタ１２２に読み込む。

【００６１】

・ レジスタＰＴＲ ＴＣ内に含まれている現在のアドレスを、レジスタＰＲＥＶ ＰＴＲ ＴＣに読み込む、

・ ポインタ・レジスタ１２２により特定されているメモリ２０′ の位置からのスレッド・コンテキストＦを、コンテキスト・レジスタ・セット１４に読み込み、スレッドＤの実行を再び開始する。

【００６２】

時刻ｔ６では、スレッドＤの実行中に、プロセッサは、プログラム・カウンタ・レジスタＰＣにより特定されているメモリ２０′ 内のそれぞれの位置から命令Ｓ／Ｗ ＩＮＴＥＲＲＵＰＴを読出す。この命令は、ハードウェア割込みのソフトウエア・エミュレーションを可能にするために与えられる。この例では、その命令は、割込みＩＮＴ １をエミュレートする。この命令に応答して、プロセッサは、ハードウェア割込みＩＮＴ １を受けた時と同じように応答する。すなわち、プロセッサは、以下の処理を行う。

【００６３】

・ コンテキスト・レジスタ・セット１４からの現在実行中のＴＨＲＥＡＤ ＣＯＮＴＥＸＴ Ｄを、レジスタＰＴＲ ＴＣに含まれているアドレスによって特定されているメモリ位置に読み込む。

【００６４】

・ レジスタＰＴＲ ＴＣに現在含まれているアドレスを、レジスタＰＲＥＶ ＰＴＲ ＴＣに読み込む。

【００６５】

・ 割込みＩＮＴ １に関連付けられているメモリ位置、すなわち、ポインタＰｔｒ ＴＣ ＩＮＴ １が保存されているメモリ位置、を特定するアドレスをスレッド・コンテキスト・ロケータ１２０に発生させる。

【００６６】

・ ＴＨＲＥＡＤ ＣＯＮＴＥＸＴ Ｆのメモリ位置を特定している、ポインタＰｔｒ ＴＣ ＩＮＴ １を、ポインタ・レジスタ１２２に読み込む。

【００６７】

・ このスレッド・コンテキストを、コンテキスト・レジスタ・セット１４に読み込み、スレッドＦの実行を開始する。

【００６８】

これまで説明してきたように、プロセッサは単一のコンテキスト・レジスタ・セット１４に含まれているコンテキストを迅速に変更する性能を有する。しかし、コンテキストを変更する速さを増すために、プロセッサは複数のコンテキスト・レジスタ・セットを有することが好ましい。その場合には、コンテキスト・キャッシュ・マネジャー１２４をプロセッサに含めることを選択できる。

【００６９】

最速化を達成するために、プロセッサにより実行されるべき種々のスレッドの数と同数のコンテキスト・レジスタ・セットが、キャッシュにある。しかし、そのような高速があらゆるスレッドに対して不要である場合には、キャッシュ・メモリ空間の使用が非効率的である。本発明の他の特徴によれば、スレッドより少ない数のコンテキスト・レジスタ・セットを有する。しかし、優先順位がより低いスレッドよりも、優先順位がより高いスレッドの方に、スレッド当りに、より多くのレジスタ・セットを使用できるようにすることによって、最適な効率が達成される。図３および図４はそのような優先順位付けの２つの例を示す。

【００７０】

図３は、キャッシュメモリ内に配置されて、キャッシュ・マネジャー１２４の制御の下にある４つのコンテキスト・レジスタ・セット（Ｉ、ＩＩ、ＩＩＩ、ＩＶ）を用いることによって、８つのスレッド（スレッドＡ、Ｂ、．．．Ｈ）を実行するように構成されている装置におけるスレッド処理を表す。各コンテキスト

・レジスタ・セットは図１に示されているそれに類似するが、８つのスレッドのおのおののコンテキストに含まれている優先順位コードｐを保持するレジスタを更に含む。この例では、

・　高い優先順位（ｐ＝１）のスレッドまたは低い優先順位（ｐ＝０）のスレッドを指示するために単一のビットが用いられる、

・　スレッドＡ、Ｂ、Ｃは高い優先順位のスレッドとして指定されている、

・　スレッドＤ、Ｅ、Ｆ、Ｇ、Ｈは低い優先順位のスレッドとして指定されている。

【００７１】

図３の例では、時刻ｔ０でプロセッサ１０は、どのスレッドも実行しておらず、コンテキスト・レジスタ・セットＩ、ＩＩ、ＩＩＩ、ＩＶは、それらのレジスタの１つもスレッド・コンテキストを含んでいない状態であるリセット状態にある。プロセッサはそれが次のようにして割込まれるまでアイドル状態にある。

【００７２】

・　時刻ｔ１では、割込みＥＸＴ　２がバス３０を介して受けられる。プロセッサは（スレッド・コンテキスト・ロケータ１２０を介して）ポインタＰｔｒ　ＴＣ　ＴＣ　ＥＸＴ　２を、ポインタ・レジスタ１２２に読み込み、その後でＴＨＲＥＡＤ　ＣＯＮＴＥＸＴ　Ｂをこのポインタにより特定されている、メモリ位置から読み出す。キャッシュ・マネジャー１２４は、このスレッド・コンテキストが高い優先順位のコードｐ＝１を含むことと、全部で４つのコンテキスト・レジスタ・セットがフリーである（すなわち、どのコンテキストも含んでいない）こと、を判定し、ＴＨＲＥＡＤ　ＣＯＮＴＥＸＴ　Ｂをコンテキスト・レジスタ・セットＩへ送る。（この例では、キャッシュ・マネジャーは最初のフリーなレジスタ・セットを数値的に常に選択するが、これは、ランダムに含んでいる、任意のシーケンスにおいて行うことができる。）コンテキスト・レジスタ・セットＩは今はアクティブとなり、プロセッサがスレッドＢを実行するにつれて連続的に更新される。

【００７３】

・　時刻ｔ２では、プロセッサ内のタイマから割込みＩＮＴ　１が受信される。

プロセッサは、ポインタＰｔｒ　ＴＣ　ＩＮＴ　１を、ポインタ・レジスタ１２２に読み込み、その後でＴＨＲＥＡＤ　ＣＯＮＴＥＸＴ　Ｆを、このポインタにより特定されているメモリ位置から読み出す。キャッシュ・マネジャーは、このスレッド・コンテキストが低い優先順位のコードｐ＝０を含んでいることと、どのコンテキスト・レジスタ・セットも低い優先順位のコンテキストを含んでいないことと、コンテキスト・レジスタ・セットＩＩ、ＩＩＩ、ＩＶはフリーであること、を判定し、ＴＨＲＥＡＤ　ＣＯＮＴＥＸＴ　Ｆをコンテキスト・レジスタ・セットＩＩへ送る。（あるいは、コンテキスト・レジスタ・セットの特定の１つを低い優先順位のスレッド・コンテキストのために保留できる。これによって、コンテキスト・レジスタ・セットのいずれも低い優先順位のスレッド・コンテキストのいずれも含んでいない、という判定ステップが無くされる。）コンテキスト・レジスタ・セットＩＩはアクティブとなり、プロセッサはスレッドＦの実行を開始する。

【００７４】

・　時刻ｔ３では、割込みＥＸＴ　１がバス３０を介して受信される。プロセッサは、ポインタＰｔｒ　ＴＣ　ＥＸＴ　１をポインタ・レジスタ１２２に読み込み、その後でＴＨＲＥＡＤ　ＣＯＮＴＥＸＴ　Ａをこのポインタにより特定されているメモリ位置から読み出す。キャッシュ・マネジャー１２４は、このスレッド・コンテキストが高い優先順位のコードｐ＝１を含んでいることと、コンテキスト・レジスタ・セットＩＩＩとＩＶがフリーであることと、を判定し、ＴＨＲＥＡＤ　ＣＯＮＴＥＸＴ　Ａをコンテキスト・レジスタ・セットＩＩＩへ送る。このレジスタ・セットはアクティブとなり、プロセッサはスレッドＡの実行を開始する。

【００７５】

・　時刻ｔ４では、スレッドＡの実行中に、プロセッサは命令ＳＷＩＴＣＨ　ＴＯ　Ｐ０　ＴＣを、現在コンテキスト・レジスタ・セットＩＩＩにあるプログラム・カウンタ・レジスタＰＣによって特定されているメモリ２０′内のそれぞれの位置から読出す。（この例では、ＴＨＲＥＡＤ　ＣＯＮＴＥＸＴ　Ｇのメモリ位置に対するアドレスがコンテキスト・レジスタ・セットＩＩＩのＰ０レジスタ

に予め保持される。また、スレッドG内に含まれている、優先順位コードｐ＝０によってキャッシュ・マネジャーが、このスレッド・コンテキストをコンテキスト・レジスタ・セットＩＩに読み込むべきであることを決定可能にされる。）この命令の実行においてはプロセッサは、コンテキスト・レジスタ・セットが１つだけ存在する場合におけるように、次のように処理する。

【００７６】

・　コンテキスト・レジスタ・セットＩＩ内のレジスタＰＴＲ　ＴＣに含まれている現在のスレッド・コンテキスト・アドレスにより特定されているメモリ位置にＴＨＲＥＡＤ　ＣＯＮＴＥＸＴ　Ｆを読み込み、コンテキスト・レジスタ・セットＩＩＩ内のレジスタＰＴＲ　ＴＣに現在含まれているアドレスをレジスタＰＲＥＶ　ＰＴＲ　ＴＣに読み込む。

【００７７】

・　コンテキスト・レジスタ・セットＩＩＩのＰ０レジスタからのＴＨＲＥＡＤ　ＣＯＮＴＥＸＴ　Ｇのアドレスを、ポインタ・レジスタ１２２に読み込む。

【００７８】

・　ＴＨＲＥＡＤ　ＣＯＮＴＥＸＴ　Ｇをコンテキスト・レジスタ・セットＩＩに読み込む。レジスタ・セットＩＩは今はアクティブとなり、プロセッサはスレッドＧの実行を開始する。

【００７９】

・　時刻ｔ５では、割込みＥＸＴ　３がパス３０を介して受信される。プロセッサはポインタＰｔｒ　ＴＣ　ＥＸＴ３をポインタ・レジスタ１２２に読み込み、その後でＴＨＲＥＡＤ　ＣＯＮＴＥＸＴ　Ｃをこのポインタにより特定されているメモリ位置から読出す。キャッシュ・マネジャー１２４は、このスレッド・コンテキストが高い優先順位のコードｐ＝１を含んでいることと、コンテキスト・レジスタ・セットＩＶのみがフリーであることと、を判定し、ＴＨＲＥＡＤ　ＣＯＮＴＥＸＴ　Ｃをコンテキスト・レジスタ・セットＩＶへ送る。このレジスタ・セットはアクティブとなり、プロセッサはスレッドＣの実行を開始する。

【００８０】

この時刻から、スレッド・コンテキストＡ、Ｂ、Ｃのそれぞれは、それぞれの

コンテキスト・レジスタ・セットＩＩＩ、Ｉ、ＩＶに保持され、それらのスレッドのそれぞれは、それぞれのレジスタ・セットをアクティブにするだけで実行できる。スレッドD、E、F、G、Hのいずれかの実行にはそれぞれのスレッド・コンテキストをコンテキスト・レジスタ・セットＩＩにロードすることを要する。

【００８１】

このように、高い優先順位のスレッドの数と少なくとも同数の専用の高い優先順位のコンテキスト・レジスタ・セット（たとえば、図３に表されている）で動作するように構成されている装置では、それらのレジスタ・セットにそれぞれのコンテキストがひとたびロードされると、それらとメモリ２０の間でコンテキストを再び転送するいかなる必要も決してない。したがって、そのような転送に付随する遅延が避けられる。逆に、低い優先順位のスレッドに対してコンテキスト・レジスタ・セットが１つだけ設けられたとすると、それらのスレッドに対するコンテキストを、プロセッサがそれらのうちの異なる１つに変更するたびに単一のレジスタ・セットへ転送しなければならない。

【００８２】

図４は、２つの優先レベルで動作する別のシステム構成にけるスレッド処理の例を表す。第１のレベルでは、図３の例におけるように、スレッド当りより多くのレジスタ・セットがより低い優先順位のスレッドよりも高い優先順位のスレッドに対して使用できるようにされる。第１の優先レベルでは、複数の専用コンテキスト・レジスタ・セットを有する各スレッド優先順位コードｐに対してあふれ優先順位アルゴリズムが利用されるが、その場合にはそれらのレジスタ・セットの数がそれぞれの優先順位コードｐを持つスレッドの数より少ない。この例では、再び、

・　高い優先順位（ｐ＝１）のスレッドまたは低い優先順位（ｐ＝０）のスレッドを指示するために単一のビットが利用される、

しかし、今では

・　スレッドA、B、C、Dは高い優先順位（すなわち、ｐ＝１）とされ、

・　割込みＥＸＴ　４に応答してプロセッサにより読出され、かつＴＨＲＥＡＤ

ＣＯＮＴＥＸＴ　Ｄのメモリ位置を有する追加のポインタＰｔｒ　ＴＣ　ＥＸＴ　４がメモリ２０′に保存されており、

・　スレッドＥ、Ｆ、Ｇ、Ｈは低い優先順位（すなわち、ｐ＝０）とされる。

【００８３】

図４の例では、時刻ｔ０から時刻ｔ５まで、動作は図３について説明したものと同じである。しかし、その後では、第２のレベルの高い優先順位のスレッドの数より、高い優先順位のコンテキスト・レジスタ・セットが少ないので、第２のレベルの優先順位アルゴリズムが実行される。たとえば、高い優先順位のスレッドが相互に等しくない優先順位のものであるとすると、第２のレベルの優先順位アルゴリズムは、そのような相対的優先順位に基づくものでもよい。あるいは、それらがすべて等しい優先順位のものであるとすると、異なるアルゴリズムを利用してもよい。以下のアルゴリズムは、高い優先順位のスレッドによるプロセッサの過去の利用の履歴に基づいている。

【００８４】

・　時刻ｔ６では、割込みＥＸＴ　４がバス３０を介して受け取られる。プロセッサはポインタＰｔｒ　ＴＣ　ＥＸＴ　４を、ポインタ・レジスタ１２２に読み込み、その後でＴＨＲＥＡＤ　ＣＯＮＴＥＸＴ　Ｄを、このポインタにより特定されているメモリ位置から読み出す。キャッシュ・マネジャー１２４は、スレッド・コンテキストが高い優先順位のコードｐ＝１を含むことと、ただしコンテキスト・レジスタ・セットはどれもフリーでないことと、を判定する。キャッシュ・マネジャーは、各スレッドに対するキャッシュ・メモリを使用するための要求の履歴をトラッキングできる。好適な実施例では、これは、各スレッドに別々のカウンタを提供し、最初にそれらのカウンタの全てを零にリセットし、その後で、要求が行われるたびに、

・要求が行われたスレッドに関連付けられているカウンタのみを零にリセットし、

・他のスレッドに対するカウンタを増加する。

【００８５】

それらのカウンタに保持されている履歴から、最後の要求がスレッドＢに対す

るものであったことをキャッシュ・マネジャーは判定する（割込みＥＸＴ　２が受信され、且つ、ＴＨＲＥＡＤ　ＣＯＮＴＥＸＴ　Ｂが、コンテキスト・レジスタ・セットＩに読み込まれた時刻ｔ１に、このスレッドＢは生じたものである）。この履歴に基づいて、プロセッサはＴＨＲＥＡＤ　ＣＯＮＴＥＸＴ　Ｂをメモリ２０′に読み込み、その後でＴＨＲＥＡＤ　ＣＯＮＴＥＸＴ　Ｄをコンテキスト・レジスタ・セットＩに読み込む。このレジスタ・セットは今はアクティブとなり、プロセッサはスレッドＤの実行を開始する。

【００８６】

多数のコンテキスト・レジスタ・セットを設けることによってプロセッサは、メモリ２０をアクセスすることなくある数のスレッドを逐次処理する。各スレッド変更に必要な時間は、最も速いバスを介してメモリをアクセスするために必要とされるはるかに長い時間ではなくて、クロック・サイクルで測定される。

【図面の簡単な説明】

【図１】

本発明の一実施例に係る多重スレッド処理装置の動作を全体的に示す図である。

【図２】

ポインタおよびスレッド・コンテキストの特定のセットがメモリに保存されている図１の多重スレッド処理装置の動作を示す図である。

【図３】

優先順位付けされたスレッドでの図１の多重スレッド処理装置の動作を示す図である。

【図４】

第１のレベルおよび第２のレベルの優先順位を有する図１の多重スレッド処理装置の動作を示す図である。

【符号の説明】

１０　プロセッサ

１２　割込み処理器

１４　コンテキスト・レジスタ・セット

１６　命令復号器

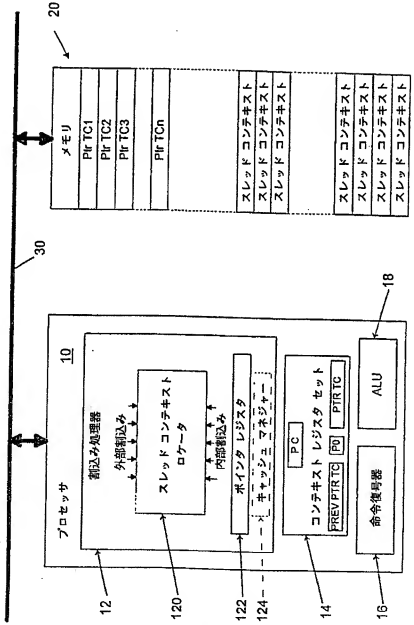
２０　メモリ

１２０　スレッド・コンテキスト・ロレータ

１２２　ポインタ・レジスタ

１２４　キャッシュ・マネジャー

【図１】



【図２】

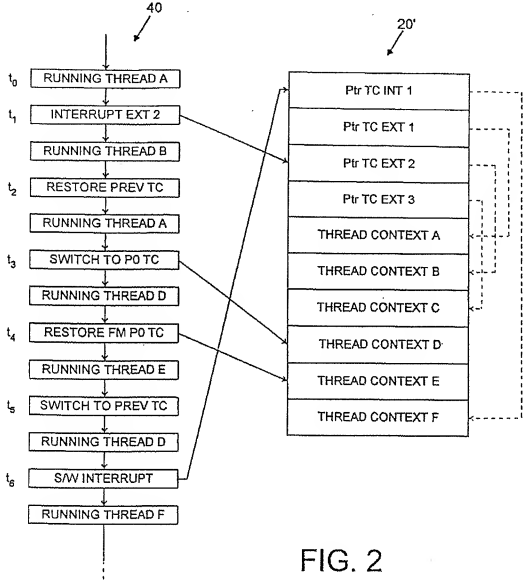

40
20'
t_0
RUNNING THREAD A
t_1
INTERRUPT EXT 2
RUNNING THREAD B
t_2
RESTORE PREV TC
RUNNING THREAD A
t_3
SWITCH TO P0 TC
RUNNING THREAD D
t_4
RESTORE FM P0 TC
RUNNING THREAD E
t_5
SWITCH TO PREV TC
RUNNING THREAD D
t_6
S/W INTERRUPT
RUNNING THREAD F
Ptr TC INT 1
Ptr TC EXT 1
Ptr TC EXT 2
Ptr TC EXT 3
THREAD CONTEXT A
THREAD CONTEXT B
THREAD CONTEXT C
THREAD CONTEXT D
THREAD CONTEXT E
THREAD CONTEXT F

FIG. 2

【図３】

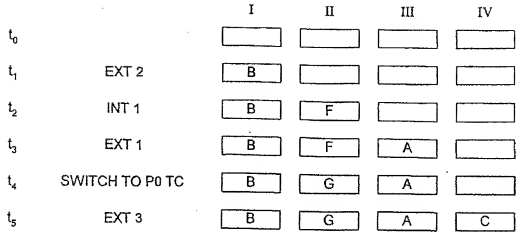


FIG. 3

【図４】

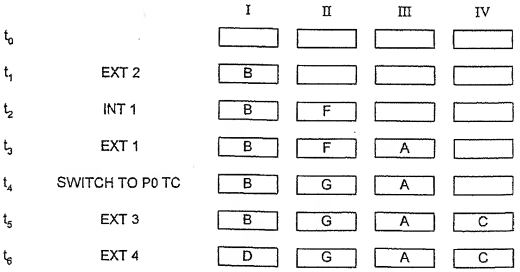


FIG. 4

【国際調査報告】

INTERNATIONAL SEARCH REPORT

Inter. nal Application No
PCT/EP 99/10170

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F9/46

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED
Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	US 4 074 353 A (IZBICKI KENNETH J ET AL) 14 February 1978 (1978-02-14) column 10, line 56 -column 12, line 38 column 13, line 49 -column 16, line 48	1,2,4, 19,30 5-9,11, 12,20-25
X	US 5 659 749 A (MITCHELL BOB ET AL) 19 August 1997 (1997-08-19) column 2, line 16 - line 51 column 5, line 14 - line 26 column 6, line 18 -column 7, line 57	1,3,4,30
Y	US 5 696 957 A (HARA KAZUHIKO ET AL) 9 December 1997 (1997-12-09) column 6, line 8 - line 23 -/—	12

X Further documents are listed in the continuation of box C.

X Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
23 June 2000	05/07/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Bijn, K

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inter -rnal Application No
PCT/EP 99/10170

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5 742 822 A (MOTOMURA MASATO) 21 April 1998 (1998-04-21) column 6, line 10 -column 10, line 5	5-9,11, 22 10
A		
Y	US 5 349 680 A (FUKUOKA KATSUHITO) 20 September 1994 (1994-09-20) column 14, line 13 -column 20, line 21	5,21
Y	US 4 410 939 A (KAWAKAMI KATSURA) 18 October 1983 (1983-10-18) column 3, line 20 - last line column 4, last line -column 6, line 37	20,23-25

1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

information on patent family members

Inter. Application No PCT/EP 99/10170

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
US 4074353	A	14-02-1978	BE	854924 A	16-09-1977
			CA	1084170 A	19-08-1980
			DE	2722099 A	08-12-1977
			FR	2353102 A	23-12-1977
			GB	1547312 A	06-06-1979
			HK	37280 A	18-07-1980
			JP	1237667 C	31-10-1984
			JP	52144243 A	01-12-1977
			JP	59011943 B	19-03-1984
US 5659749	A	19-08-1997	NONE		
US 5696957	A	09-12-1997	JP	5265753 A	15-10-1993
US 5742822	A	21-04-1998	JP	8171494 A	02-07-1996
			GB	2296354 A,B	26-06-1996
US 5349680	A	20-09-1994	JP	4172551 A	19-06-1992
			JP	2062324 C	24-06-1996
			JP	4182834 A	30-06-1992
			JP	7095276 B	11-10-1995
			KR	9512293 B	16-10-1995
US 4410939	A	18-10-1983	JP	56016248 A	17-02-1981

Form PCT/ISA/210 (patent family annex) (July 1992)

フロントページの続き

(81)指定国 EP(AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), CN, JP, KR, SG
(71)出願人 Groenewoudseweg 1, 5621 BA Eindhoven, The Netherlands
Fターム(参考) 5B098 AA10 BA06 BA12 CC02 GA05 GC03